What is claimed is:

1. A method for managing integrity of a file, the method comprising:

a) at a first time, performing a content checksum of a file in a first repository node to obtain a first checksum and storing the first checksum in the first repository node;

b) at a second time, re-performing the content checksum on the file to obtain a second checksum and comparing the second checksum with the first checksum; and

c) if the second checksum does not equal the first checksum, then at a node with a replica, verifying the replica;

if the replica is verified, then transmitting the verified replica to the first repository node; and

replacing the file with the verified replica;

if the replica is not verified, determining if all other repository nodes with replicas have been checked;

if not, then selecting a node with an alternative replica that has not been verified and verifying the alternative replica;

if the alternative replica is verified, then transmitting the verified alternative replica to the first repository node; and

replacing the file with the verified alternative replica;

if all other repository nodes with replicas have been checked and no verified replicas have been discovered, then determining that file integrity correction failed.

2. The method of claim 1 wherein the file is a first file and wherein the method further comprises:

if at least one of a file, a replica, and an alternative replica is verified, then determining if all relevant files have received integrity management;

if all relevant files have received integrity management, then determining that integrity management is complete;

if not, then recursively setting the file equal to the next file and performing (a) -(c) on the next file until all relevant files have received integrity management.

3. The method of claim 1 wherein verifying the replica comprises performing a content checksum on the replica to obtain a replica checksum and determining whether the replica checksum equals the first checksum.

4. A method for managing integrity of a file, the method comprising:

a) at a first time, performing a content checksum of a file in a first repository node to obtain a first checksum and storing the first checksum in the first repository node;

b) at a second time, re-performing the content checksum on the file to obtain a second checksum and comparing the second checksum with the first checksum; and

c) if the second checksum does not equal the first checksum, then recovering the file from another node.

5. The method of claim 4 wherein recovering the file from another node comprises:

determining repository nodes that have a replica of the file; and

querying the repository nodes..

6. The method of claim 5 wherein recovering the file from another node further comprises:

at a node with a replica, verifying the replica;

if the replica is verified, then transmitting the verified replica to the first repository node; and

replacing the file with the verified replica.

7. The method of claim 6 wherein verifying the replica comprises performing a content checksum on the replica to obtain a replica checksum and determining whether the replica checksum equals the first checksum.

8. The method of claim 6 wherein recovering the file from another node further comprises:

if the replica is not verified, determining if all other repository nodes with replicas have been checked;

if not, then selecting a node with an alternative replica that has not been verified and verifying the alternative replica;

if the alternative replica is verified, then transmitting the verified alternative replica to the first repository node; and

replacing the file with the verified alternative replica.

9. The method of claim 8 wherein the method further comprises:

if all other repository nodes with replicas have been checked and no verified replicas have been discovered, then determining that file integrity correction failed.

10. The method of claim 8 wherein the file is a first file and wherein the method further comprises:

if at least one of a file, a replica, and an alternative replica is verified, then determining if all relevant files have received integrity management;

if all relevant files have received integrity management, then determining that integrity management is complete;

if not, then recursively setting the file equal to the next file and performing (a) -(c) on the next file until all relevant files have received integrity management.

11. The method of claim 4 wherein the checksum is an MD5 checksum.

12. A data protection system comprising:

a primary repository node having:

a data mover operative to manage the transfer of data;

a primary repository API in communication with the data mover and operative to communicate with a network;

a primary repository file transfer module in communication with the data mover and operative to receive files;

an integrity service operative:

at a first time, to perform a content checksum of a file in a first repository node to obtain a first checksum and to store the first checksum in the primary repository node, and

at a second time, to re-perform the content checksum on the file to obtain a second checksum and to compare the second checksum with the first checksum, and

if the second checksum does not equal the first checksum, to output a file recovery request; and

a replicator service in communication with the data mover and the integrity service, the replicator service operative to

receive the recovery request from the integrity service and to manage the process of recovering the file from another node.